BAB 4
IMPLEMENTASI ALGORITMA DAN HASIL PENGUJIAN

4.1 Implementasi

Tahap implementasi dilakukan setelah proses analisis selesai dilakukan selanjutnya diimplementasikan ke dalam bahasa pemrograman. Setelah implementasi algoritma dilakukan maka dilakukan pengujian terhadap algoritma yang digunakan pada aplikasi yang dibuat dengan tujuan untuk mengetahui apakah algoritma dan aplikasi yang dibuat dapat berjalan sesuai dengan tujuan atau tidak.

4.1.1 Implementasi Algoritma

Pada implementasi algoritma menjelaskan implementasi dan melakukan pengujian algoritma greedy dan logika fuzzy terhadap prioritas penyerangan pada *game* RPG Me vs Zobie serta menjelaskan skenario dari hasil pengujian algoritma logika fuzzy yang diterapkan pada *game* bertipe RPG.

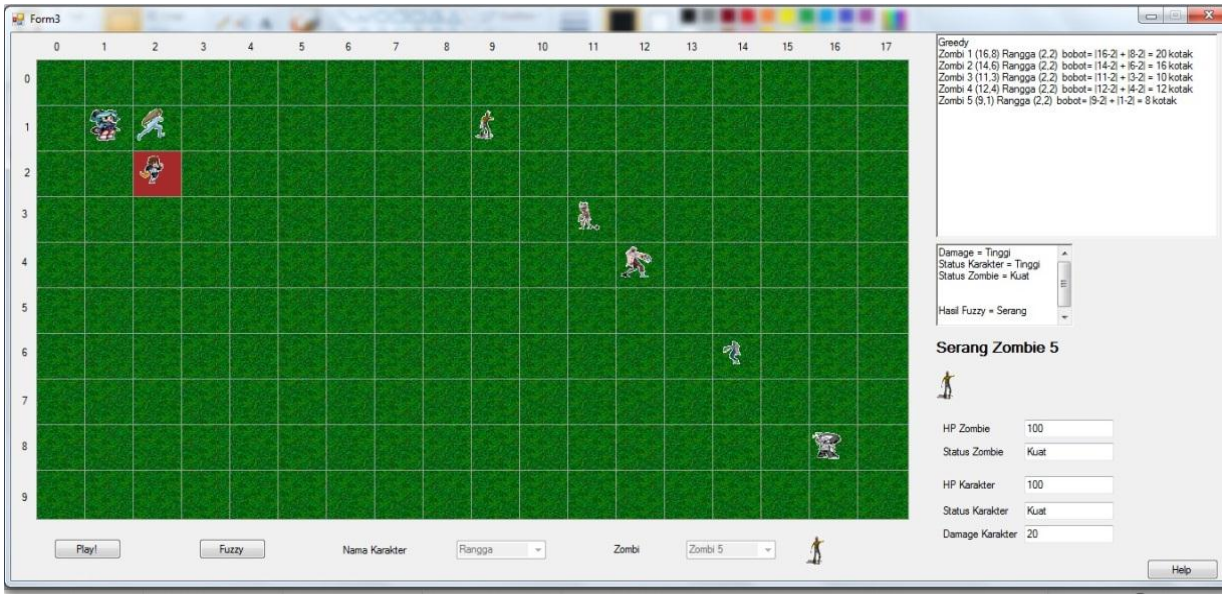
4.1.1.1 Pengujian Algoritma dan Skenario Pertama

1. Karakter Mencari rute Terpendek

Karakter yang memiliki giliran pertama akan mencari rute terpendek antar karakter dan musuh yang dituju. Karakter yang mempunyai giliran menyerang berada pada kotak warna merah. Jarak karakter dan musuh dilihat berdasarkan titik koordinat antara karakter dan musuh. Berikut titik koordinat karakter dan musuh dapat dilihat pada tabel 4.1

Tabel 4.1 Table Titik Koordinat Karakter dan Musuh

No	Nama Karakter	Koordinat karakter	Nama Musuh	Koordinat Musuh
1	Rangga	(2,2)	Zombie 1	(16,8)
			Zombie 2	(14,6)
			Zombie 3	(11,3)
			Zombie 4	(12,4)
			Zombie 5	(9,1)



Gambar 4.1 Posisi Dan Jarak Karakter ke Musuh Skenario Pertama

Skenario Pengujian

Player memilih karakter, kemudian *player* memilih *zombie* 5 yang berada pada koordinat (9,1).

Output Algoritma Greedy:

1. Koordinat karakter (2,2), Koordinat zombie 1 (16,8) = 20 kotak
2. Koordinat karakter (2,2), Koordinat zombie 2 (14,6) = 16 kotak

3. Koordinat karakter (2,2), Koordinat zombie 3 (11,3) = 10 kotak
4. Koordinat karakter (2,2), Koordinat zombie 4 (12,4) = 12 kotak
5. Koordinat karakter (2,2), Koordinat zombie 5 (9,1) = 8 kotak

Maka dilihat dari hasil output dari algoritma greedy rute terpendek yang harus ditempuh oleh karakter ke musuh yaitu 8 kotak.

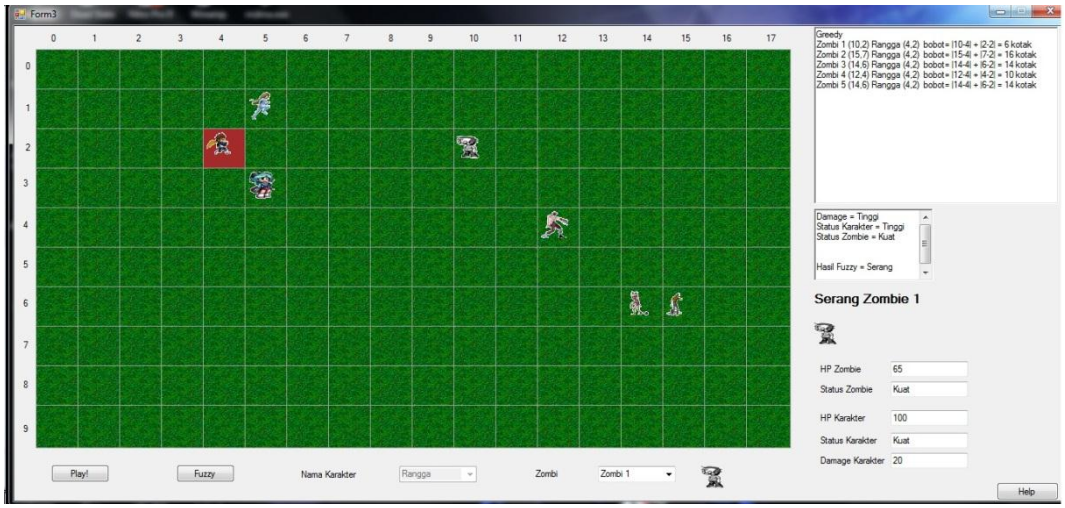
4.1.1.2 Pengujian Algoritma dan Skenario Kedua

1. Karakter Menyerang Musuh

Posisi player dan musuh akan menjadi prioritas pertama dan prioritas kedua diambil dari nilai parameter yang dimiliki oleh karakter dan musuh. Berikut nilai parameter yang dimiliki oleh musuh dan karakter pada tabel 4.2

Tabel 4.2 Tabel Parameter Karakter dan Musuh

No	Nama Karakter	Parameter Krakter	Nama Musuh	Parameter Musuh
1	Rangga	Damage : 20 Nyawa : 100	Zombie 5	Nyawa 65



Gambar 4.2 Karakter Menyerang Musuh Skenario ke-2

2. Skenario Pengujian

Player memilih karakter, kemudian memilih musuh sesuai saran yang diberikan sebagai prioritas serangan ke musuh. Kemudian player melakukan penyerangan ke mush yang dituju.

Output logika Fuzzy

1. Damage (Tinggi 20) = 0, Nyawa Karakter(Kuat, 100)= 0, Nyawa Musuh (Kuat,65) = 0,38

   Z=100

   Dilihat dari output logika fuzzy maka prioritas yang didapat oleh karakter adalah Menyerang musuh yang dituju.

4.2 Pengujuan White-Box

Pengujian white-box merupakan pengujian yang didasarkan pada pengecekan terhadap detail perancangan dengan menggunakan struktur kontrol dari desain program secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Tujuan dari pengujian white-box untuk mengetahui cara kerja suatu aplikasi aau perangkat luak secara internal dan menjamin operasi-operasi internalsesuai dengan spesifikasi yang telah ditetapkan dengan menggunakanstruktur kontrol dari prosedur yang di rancang. Berikut adalah pengujian white-box yang dilakukan pada Proses Greedy dan logika fuzzy:

a. Pengujian *white-box* terhadap Proses Greedy

```
1  private int greedy(int xx, int yy)
    {

     bobot = new int[zz];

2  for (int i = 0; i < zs; i++)
                {
                    if (zombie[i, 1] != "Mati")
                    {

3  bobot[i] = Math.Abs(xx - posz[i, 0]) + Math.Abs(yy - posz[i, 1]);

4  int min = Array.IndexOf(bobot, bobot.Min());

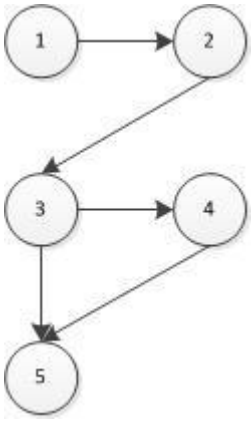
                richTextBox1.Clear();
                richTextBox1.Text += "Greedy\n";

5  }
    return min;

   }
```

Gambar 4.3 *Pseudocode* Proses Greedy

Berdasarkan *pseudocode* terhadap proses greedy di atas maka diagram alir gambar 4.4



Gambar 4.4 Diagram Alir *Pseudocode* Proses Greedy

Berdasarkan digram alir *pseudocode* terhadap proses greedy maka dapa dihitung cyclometic comlexity, yaitu:

1. Flowgrap mempunyai 1 region
2. V(G) = 5 edge -5 node + 2 = 2
3. V(G) = 1 Predicate + 1 = 2

Dari hasil perhitungan cyclometic complexity terdapat 2 independent path yaitu:

Path 1 : 1 – 2 – 3 – 4 – 5

Path 2 : 1 – 2 – 3 – 5

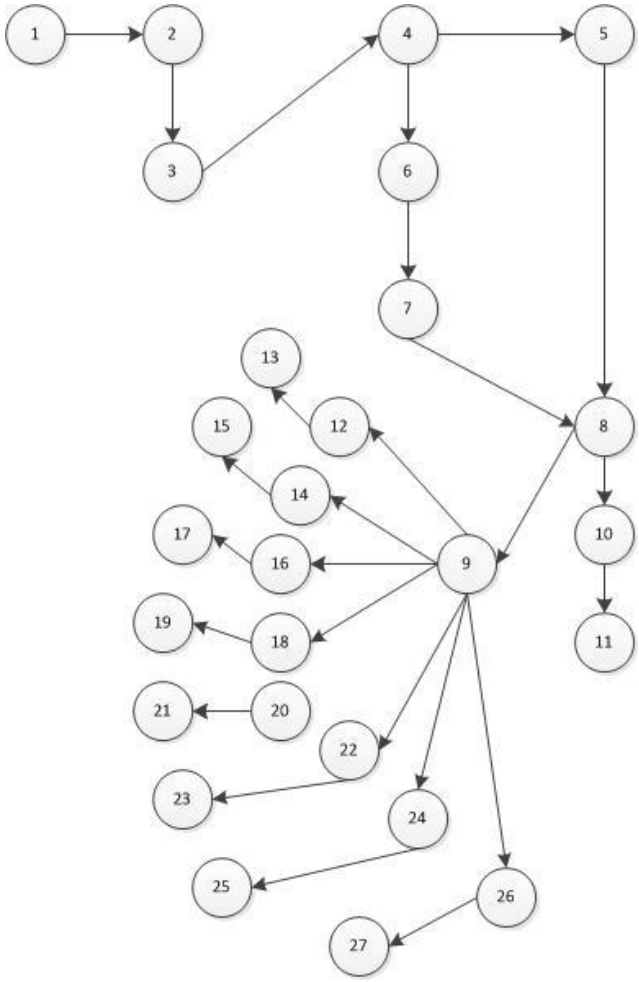
b. Logika Fuzzy

1	Input Damage
2	Imput Nyawa
3	Input Status
4	If Damage < 10
5	Kabur
6	Else
7	Serang
8	If nyawa >= 60
9	Karakter = Tinggi
10	Else
11	Karakter = Rendah
12	Rule 1

13	Hasil = serang
14	Rule 2
15	Hasil = serang
16	Rule 3
17	Hasil = kabur
18	Rule 4
19	Hasil = serang
20	Rule 5
21	Hasil = serang
22	Rule 6
23	Hasil = serang
24	Rule 7
25	Hasil = serang
26	Rue 8
27	Hasil = kabur

Gambar 4.5 *Pseudocode Fuzzy*

Berdasarkan *pseudocode* terhadap fuzzy di atas maka diagram alirnya pada gambar 4.6



Gambar 4.6 Diagram Alir *Pseudocode* Fuzzy

Berdasarkan diagram alis pseudocode terhadap fuzzy maka dapat di hitung *cyclometic complexity*, yaitu:

1. Flowgrap mempunyai1 region.
2. V(G) = 26 edge – 27 node + 2 = 1
3. V(G) = 4 predicate + 1 = 5

Dari hasil perhitungan *cyclometic complexity* terdapat 2 independent path yaitu:

Path 1 : 1 – 2 – 3

Path 2 : 4 – 5 – 6 – 7

Path 3 : 4 – 6 – 7 – 8 – 10

Path 4 : 4 – 5 – 8 – 10 -11

Path 5 : 8 – 9 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 26 - 27